黑马面经

技术点：vue2+router+sass+eslint-stardard+vuex+elementUI+axios

接口文档：<https://www.apifox.cn/apidoc/project-934563/api-19465917>

接口基地址：<http://interview-api-t.itheima.net/>

接口演示：<http://xpzll.gitee.io/hmmj-pc>

elementUI内部就是用的scss写的,所以用scss。

这是自己搭架子.vue-cli

vue创建项目：vue create 项目名

运行vue项目脚手架： npm run serve

预处理器：scss/sass(stylus)反人类。css,less

框架+组件库

elementUI网站过时，语法注意下

scss语法$声明变量。---保持多个元素同一个颜色：声明变量 $color:#f00

1. 框架搭建 vue+element 确认技术栈
2. 主题色
3. 封装axios ---设置基地址，请求拦截和响应拦截
4. 本地储存-封装操作token的三个方法：存、取、删

上面是项目准备，一般是项目经理和组长做的搭建项目和大概的架子，没有就才是组员做，

1. 页面组件-位置分析及其页面准备然后进行路由配置（1级和2级也就是登录页和骨架页layout 二级就是数据看板和面经管理）
2. 登录页面完成校验然后封装请求-用户请求

axios不直接设置基地址 axios.defaults.baseURL=’’

因为后面项目有多个基地址。Import axios from ‘axios’然后const request=axios.create({baseUrl:”,timeout:”]) 然后设置请求拦截和响应拦截。......., 最后暴露出去export default promise

另外：我发请求没反应，如果没有设置超时，大概会等1分钟

存储不习惯用这个，但是后面获取需要这个：const KEY='my-token-element-pc'

存储需要值token传入然后键值对（KEY,value）进行存储，其他不能传值，直接根据键

获取移除。目的：因为在不同页面存储取，所以封装const KEY=这样子方便后期修改

路由配置记得设置路由出口，每个页面最好设置一个名字方便后期排错

Index.vue就代表w这个页面的主页面

路由配置：组件要导入，// 名字-组件-路径，儿子children，{ name: login, path: '/login', component: login }

封装思想：**封装然后暴露出去**

**样式：background:是集合，包含图片位置大小 background:url(‘’) no-repeat center/cover , height:100%vh是高度完全覆盖，然后**

**垂直和水平居中 父级display:flex justiy-content:center align-item:center**

使用 scoped后，父组件的样式将不会渗透到子组件中(组件样式不会成为全局样式造成样式冲突，那一定要加scoped，但是加了scoped样式就作用不到card组件内部浏览器才能看到的内部标签上，所以这时候就需要深度作用选择器)

**深度作用选择器：默认情况下，如果你加了scoped，代表这个样式只能给当前组件写的标签用，也就是某个标签是组件内部的标签（只在浏览器查结构看到的那种）是无法作用到的，要想要样式作用到组件内部，就需要用深度作用选择器**

**语法： ：：v-deep**

**也就是我写的样式和组件里面的潜在样式，加了scoped，组件内部的潜在样式作用不到，就需要在：：v-deep 。作用到组件内部。不专业的叫法叫做样式穿透。Vue官网里面有说法。Scoped-css. 另外一种用法》》》 只能用在原生css里面，不能scss这种非原生预处理器**



垂直居中：line-height:80px ，font-size,color

表单：model这个属性很重要，它跟表单绑定的对象，要想要拿到表单的值，必须先给form加一个model

trigger: 'change' ：页面输入就触发

组件库知识不用背下来，直接看文档学习。框架Vue知识点还是要记忆。

整个表单校验：

原生找dom是documeng,queryselectors(‘选择器’)，首先代码比较长，无法找组件对象。其次它找到的是整个html的对象元素（明明.form是el-form组件，但是它找到的不是组件而是一个dom元素），因为只是想要一部分东西不是全部。



组件库elementUI里面找元素，

找元素然后表单验证

Vue推荐找元素的方式就是ref搭配$refs：要找一个元素就不要用elementUI内置的一些东西，而是用ref，ref起一个名字，用法:标签要找的元素：ref=’别名’ 方法里面：this.$refs.form。里面有组件内置的一些方法和一些属性。加this因为在js里面。ref既可以找dom元素也可以找组件对象，而且找组件内置的一些东西属性方法和数据，写起来比较方便。

要让整个表单校验，首先要找个这个表单对象，然后调用它的validate方法即可做整个表单校验。

elementUI官网没写这个写法：

 {

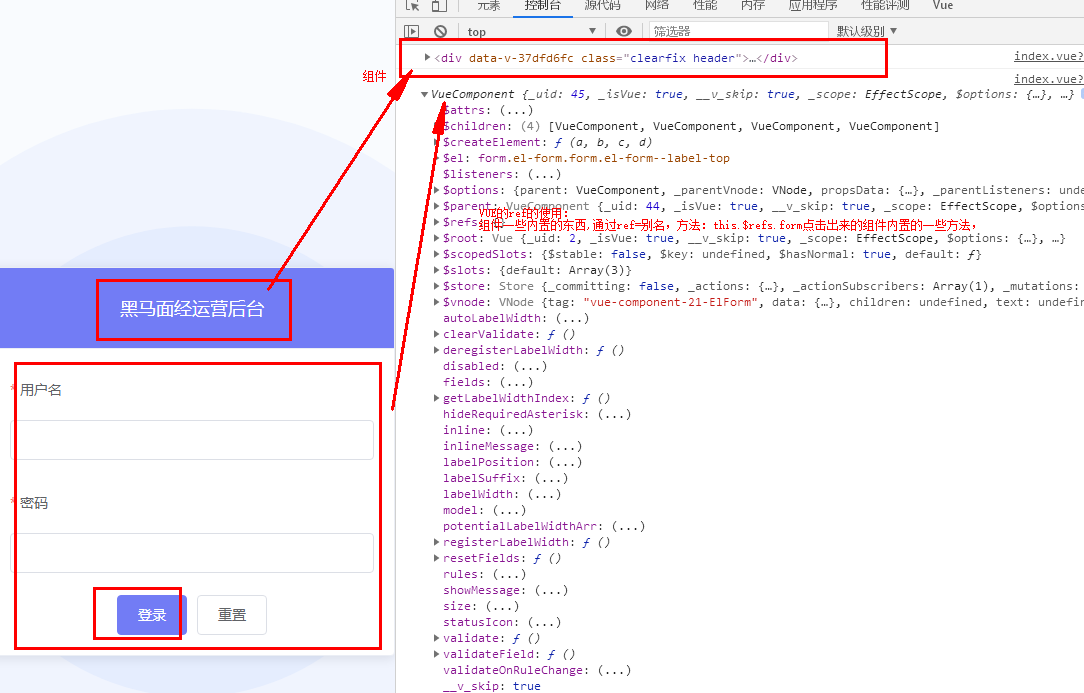
            min: 5,

            max: 11,

            message: '长度在 5 到 11 个字符',

            trigger: ['change', 'blue']

          }





重置就是做两件事：1、把数据恢复到默认值2、清空表单校验结果

封装请求：1、导入基地址2、封装一个用来登录的接口暴露出去在api文件夹里面封装user.js

账号和密码:admin8 密码admin

本地储存token：

vuex的特点：数据是响应式的,（数据改变页面也跟着改变），vue里面直接使用Vuex，本地储存要再建个变量才行，缺点就是一刷新就恢复到默认值,vuex恢复到默认值，本地储存就按储存的显示

本地储存缺点：数据不是响应式的，，在vuex的组件里无法直接使用。改变本地储存页面没有发生改变。优点：数据长久保存，只要自己不删除就一直都在。

本地储存+Vuex组合使用：1、如果一个项目一个数据多个组件使用，最好存到vuex里面，2、另外这个数据下次打开或者刷新之后还要在，那就利用本地储存+vuex

   <p>{{ $store.state.count }}</p>

    <p>{{ count }}</p>

应用：token要长期保存，另外需要在其他组件长期使用。根据之前学的模块化思想，Vuex的数据最好不要放到根store里面，最好放到模块里面命名

vuex忘记了模块化的使用



本地存储到store里面的user.js

 token: getToken() || ''

一刷新vuex就没有了，所以就需要在state里面直接获取本地存储变为默认值或者空值未登录。